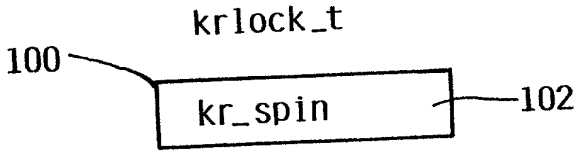
krlock_t
100
kr_spin
102

FIG. 1

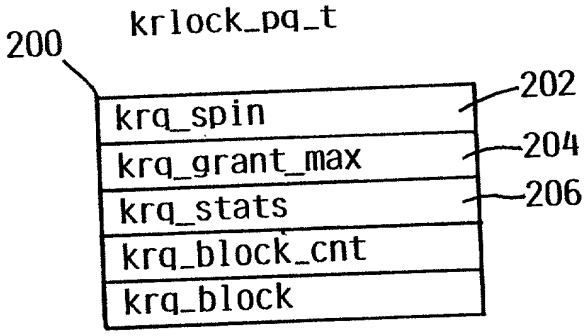
krlock_pq_t
200
krq_spin
krq_grant_max
krq_stats
krq_block_cnt
krq_block
202
204
206

FIG. 2

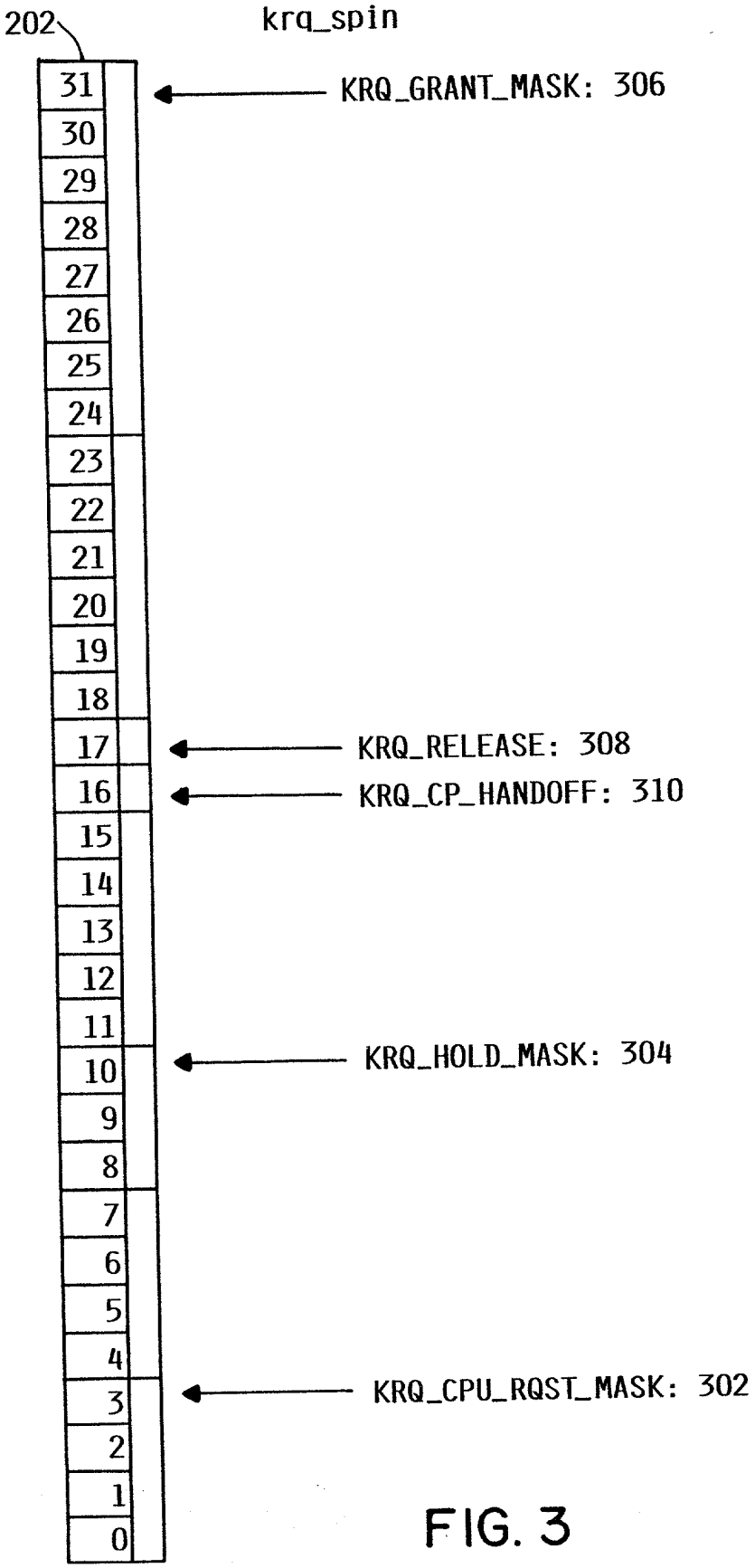
202
krq_spin
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
KRQ_GRANT_MASK: 306
KRQ_RELEASE: 308
KRQ_CP_HANDOFF: 310
KRQ_HOLD_MASK: 304
KRQ_CPU_RQST_MASK: 302

FIG. 3

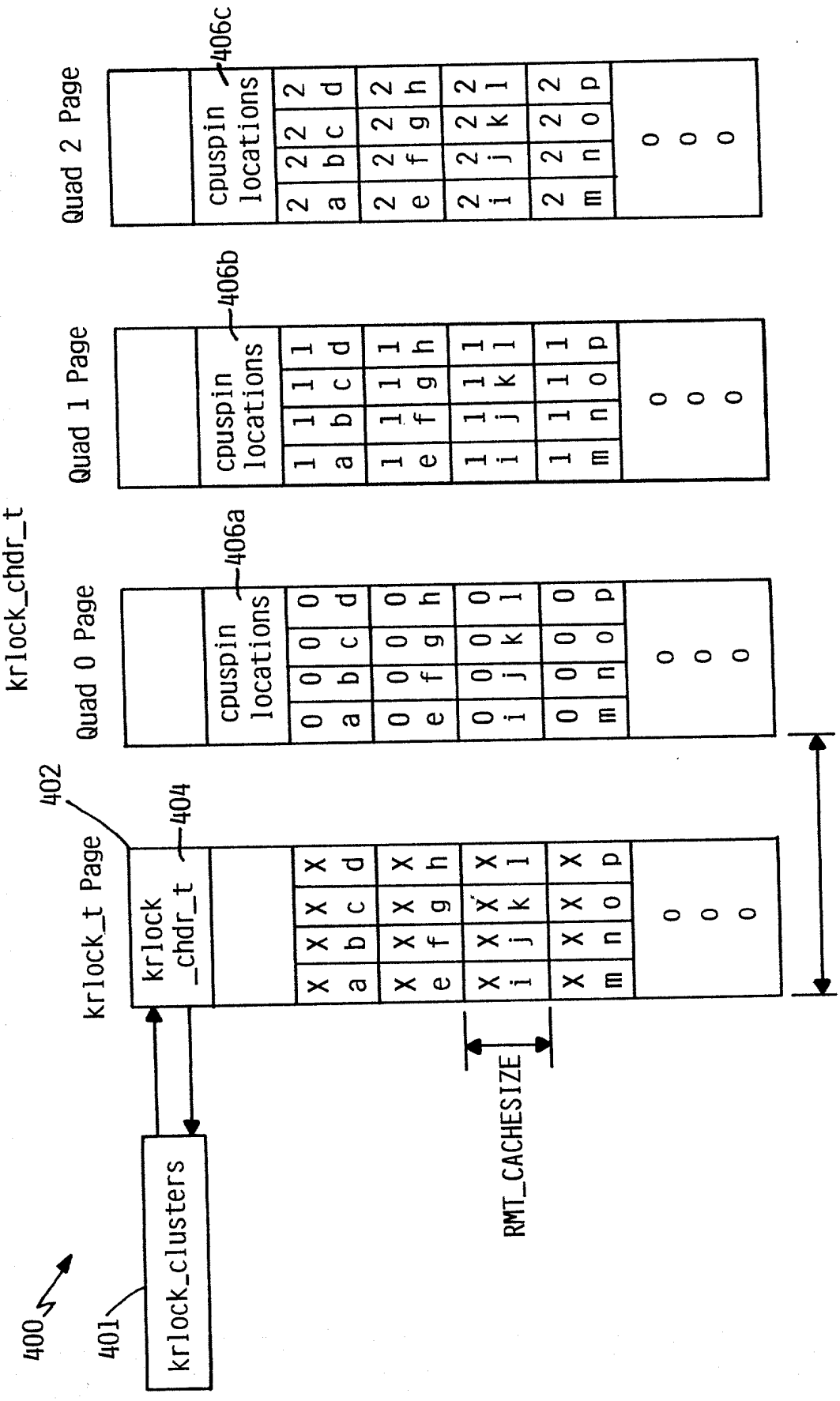
400
401
krlock_clusters
krlock_t Page
402
krlock
_chdr_t
404
krlock_chdr_t
Quad 0 Page
406a
Quad 1 Page
406b
Quad 2 Page
406c
cpuspin
locations
RMT_CACHESIZE
KRLOCK_STRIDE (Page Size)

FIG. 4

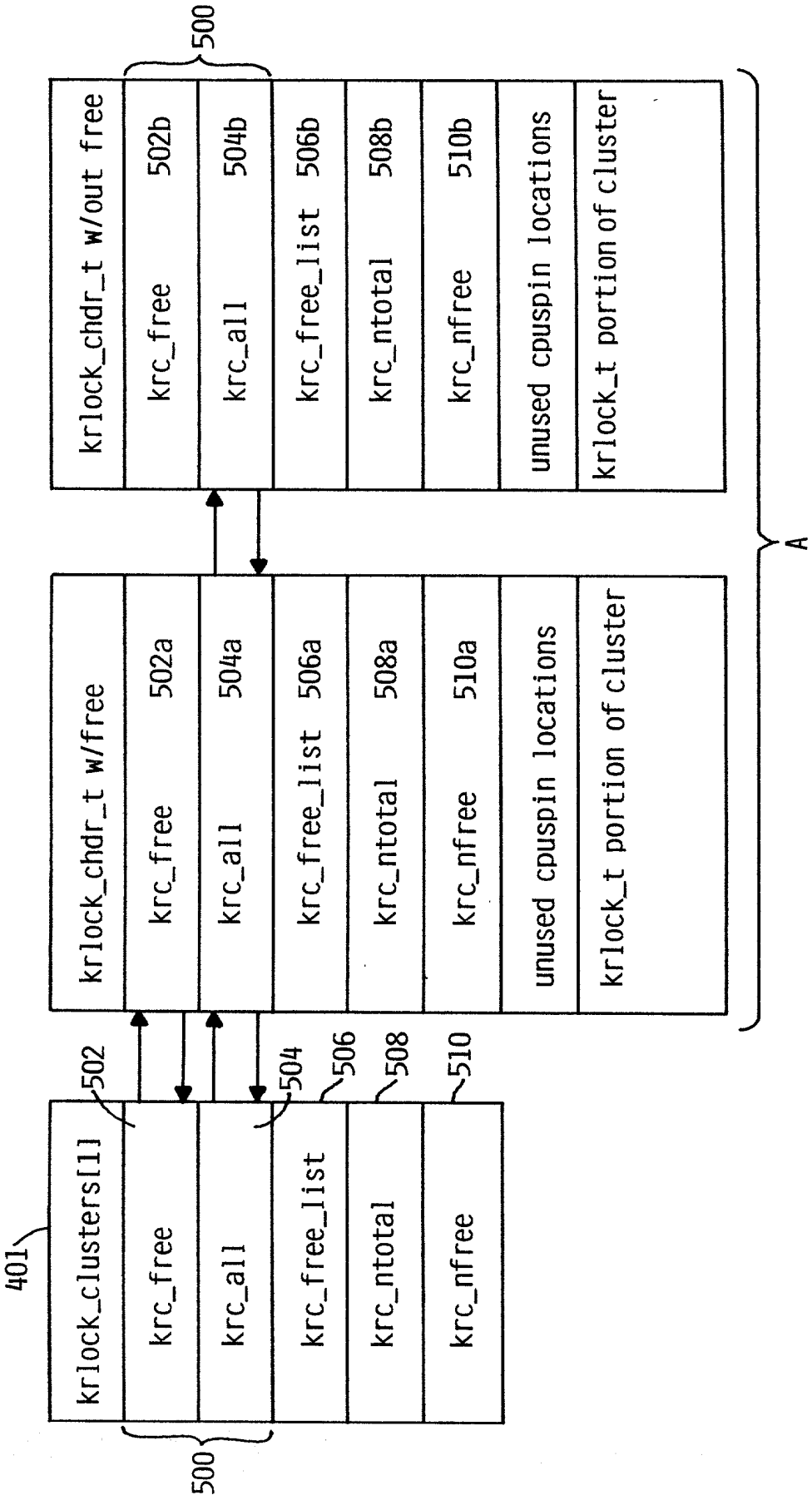
krlock_clusters
krlock_clusters[1]
krc_free
krc_all
krc_free_list
krc_ntotal
krc_nfree
401
500
502
504
506
508
510
krlock_chdr_t w/free
502a
504a
506a
508a
510a
unused cpuspin locations
krlock_t portion of cluster
krlock_chdr_t w/out free
502b
504b
506b
508b
510b
unused cpuspin locations
krlock_t portion of cluster
A

FIG. 5A

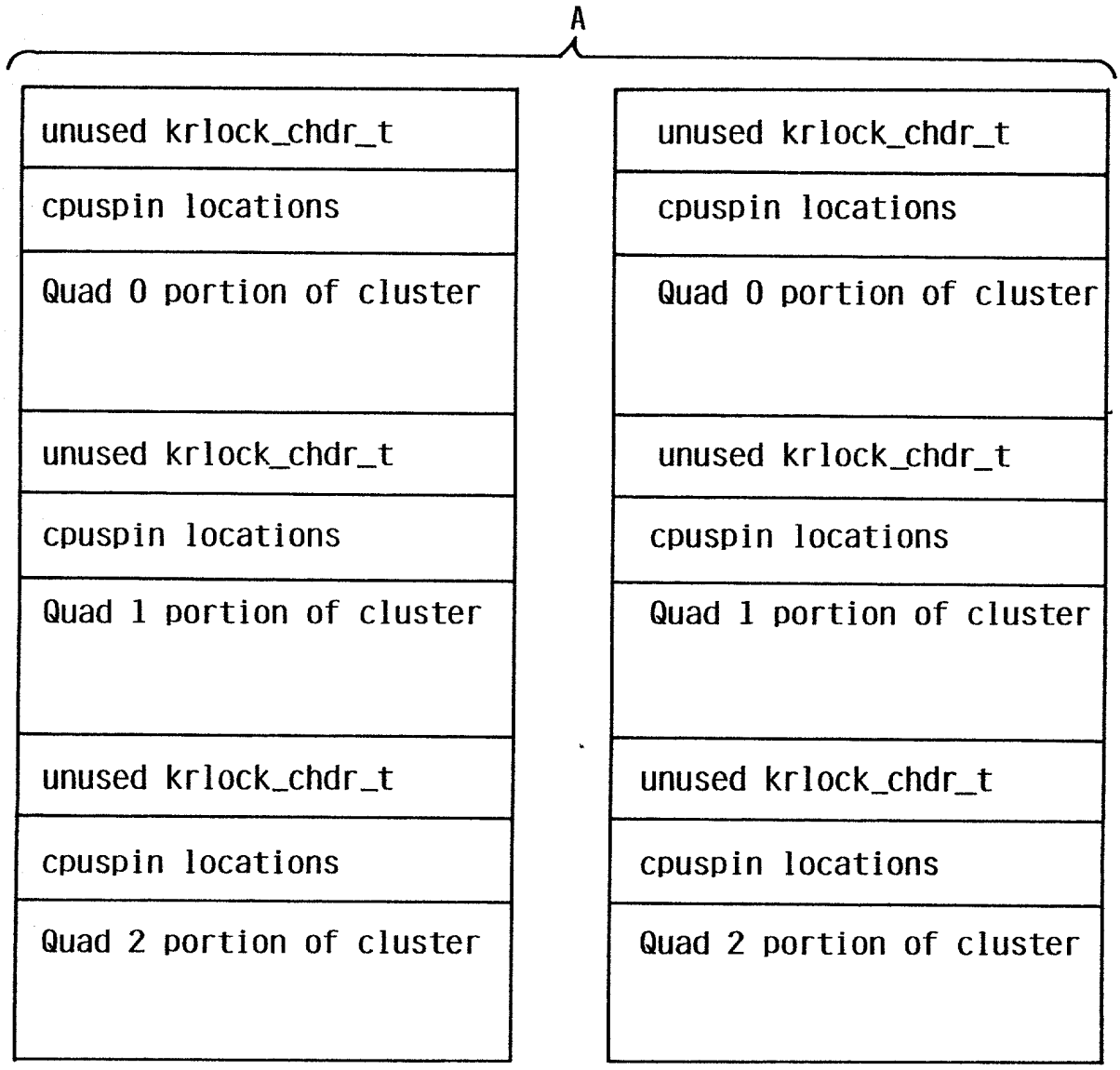
A
unused krlock_chdr_t
cpuspin locations
Quad 0 portion of cluster
unused krlock_chdr_t
cpuspin locations
Quad 1 portion of cluster
unused krlock_chdr_t
cpuspin locations
Quad 2 portion of cluster
unused krlock_chdr_t
cpuspin locations
Quad 0 portion of cluster
unused krlock_chdr_t
cpuspin locations
Quad 1 portion of cluster
unused krlock_chdr_t
cpuspin locations
Quad 2 portion of cluster

FIG. 5B

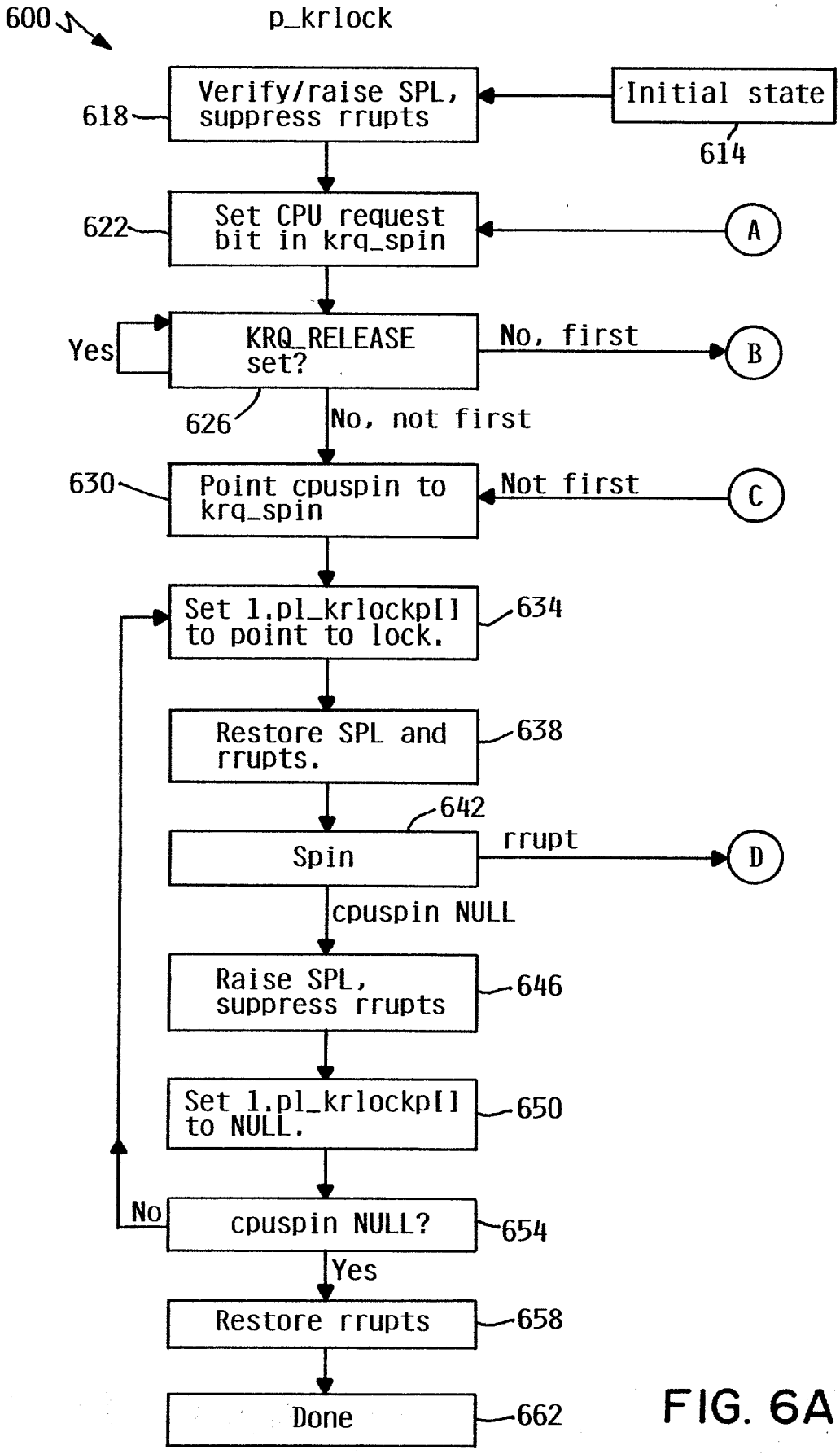
600
p_krlock
Verify/raise SPL, suppress rrupts
618
Initial state
614
Set CPU request bit in krq_spin
622
A
KRQ_RELEASE set?
Yes
No, first
B
626
No, not first
Point cpuspin to krq_spin
630
Not first
C
Set 1.pl_krlockp[] to point to lock.
634
Restore SPL and rrupts.
638
642
Spin
rrupt
D
cpuspin NULL
Raise SPL, suppress rrupts
646
Set 1.pl_krlockp[] to NULL.
650
No
cpuspin NULL?
654
Yes
Restore rrupts
658
Done
662

FIG. 6A

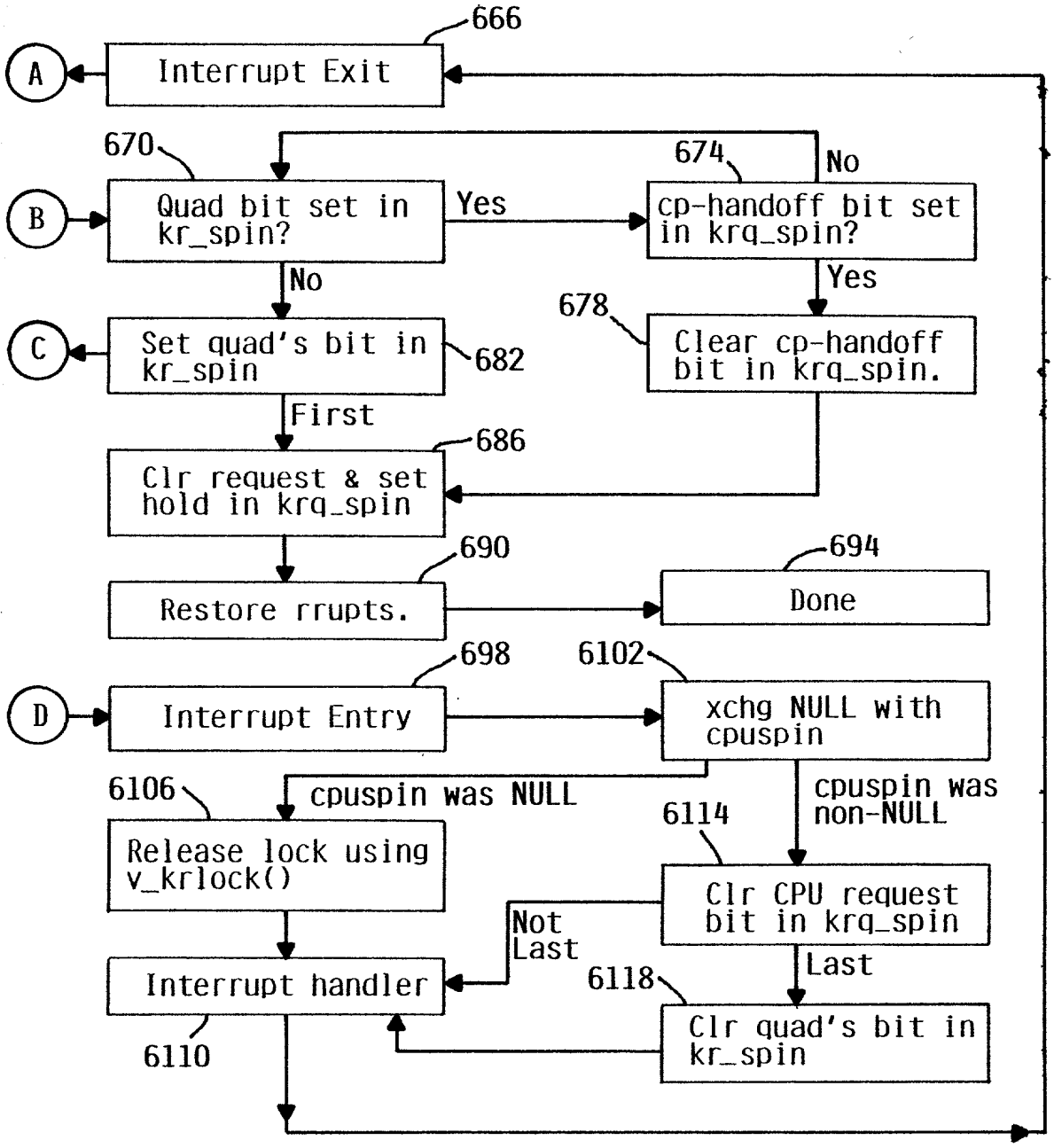
666
Interrupt Exit
A
670
Quad bit set in kr_spin?
B
Yes
674
cp-handoff bit set in krq_spin?
No
Yes
No
678
Clear cp-handoff bit in krq_spin.
C
Set quad's bit in kr_spin
682
First
686
Clr request & set hold in krq_spin
690
Restore rrupts.
694
Done
698
Interrupt Entry
D
6102
xchg NULL with cpuspin
6106
cpuspin was NULL
cpuspin was non-NULL
6114
Release lock using v_krlock()
Clr CPU request bit in krq_spin
Not Last
Last
Interrupt handler
6118
Clr quad's bit in kr_spin
6110

FIG. 6B

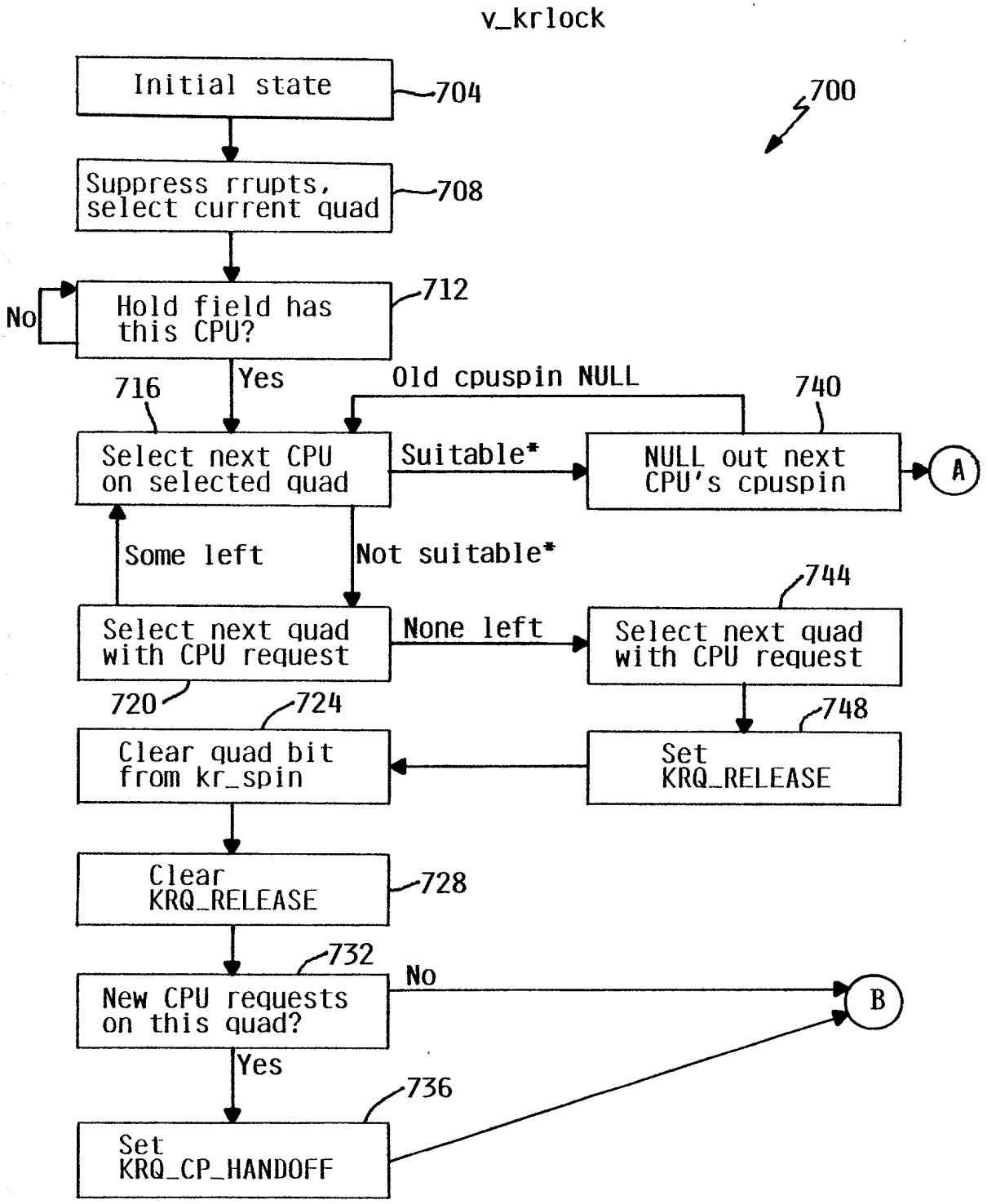
v_krlock
Initial state
704
700
Suppress rrupts, select current quad
708
No
Hold field has this CPU?
712
716
Yes
Old cpuspin NULL
740
Select next CPU on selected quad
Suitable*
NULL out next CPU's cpuspin
A
Some left
Not suitable*
744
Select next quad with CPU request
None left
Select next quad with CPU request
720
724
748
Clear quad bit from kr_spin
Set KRQ_RELEASE
Clear KRQ_RELEASE
728
732
New CPU requests on this quad?
No
B
Yes
736
Set KRQ_CP_HANDOFF

FIG. 7A

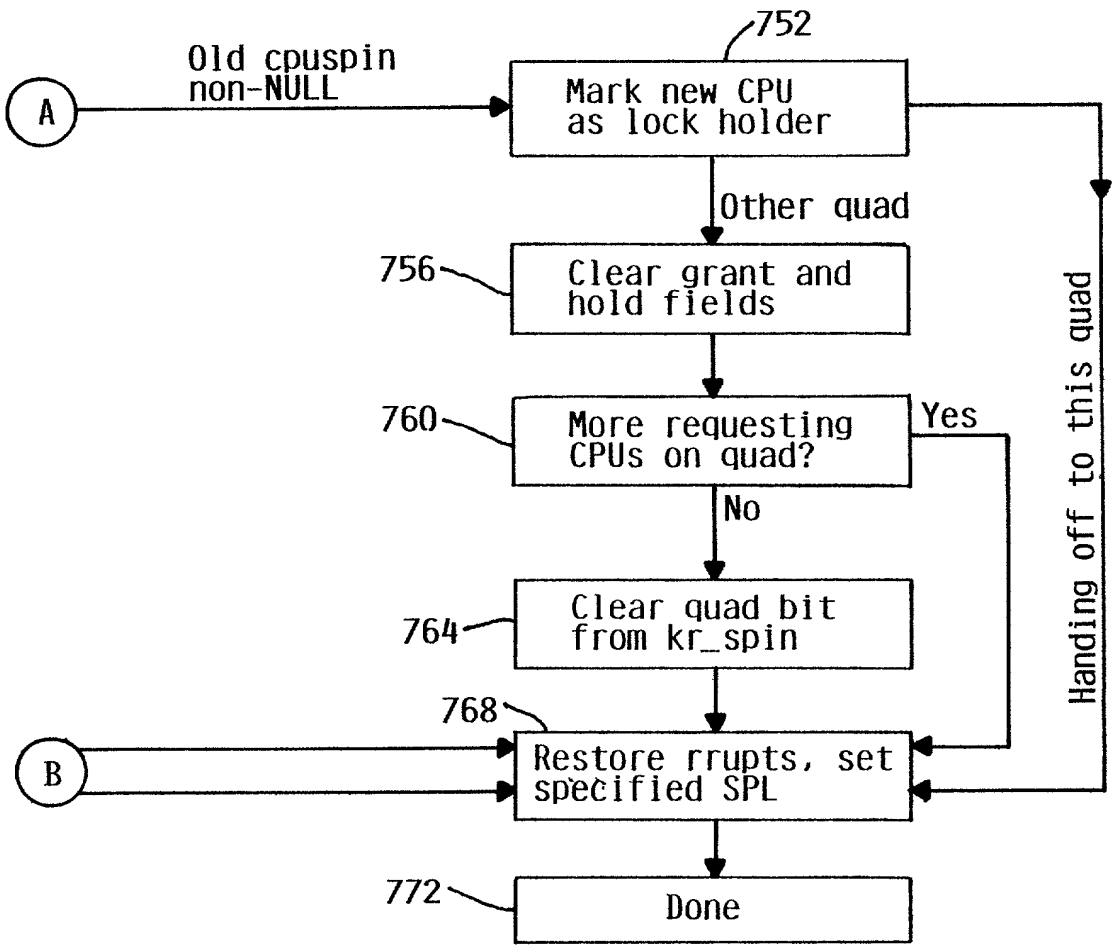
752
Old cpuspin non-NULL
A
Mark new CPU as lock holder
Other quad
756
Clear grant and hold fields
760
More requesting CPUs on quad?
Yes
No
Handing off to this quad
764
Clear quad bit from kr_spin
768
B
Restore rrupts, set specified SPL
772
Done

FIG. 7B

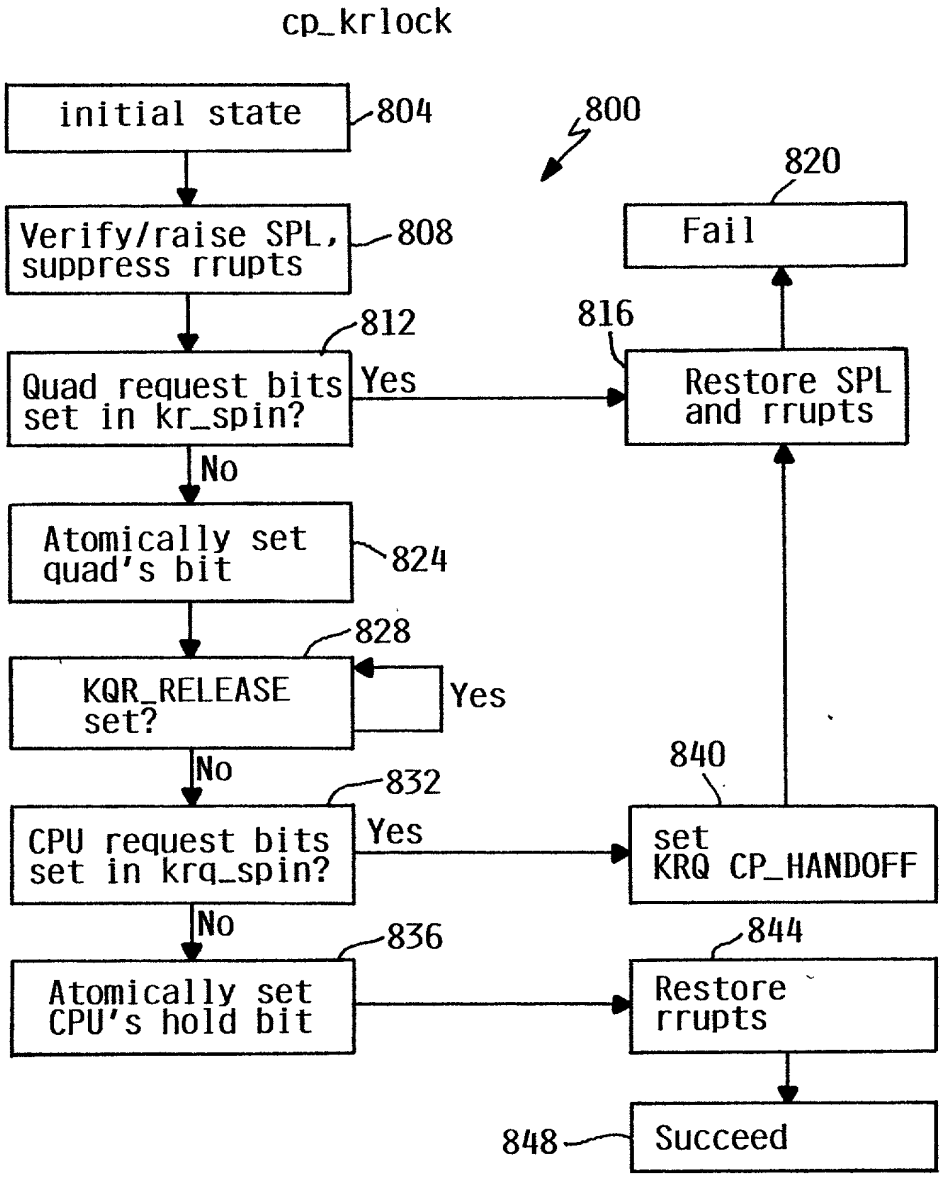
cp_krlock
initial state
804
800
820
Verify/raise SPL, suppress rrupts
808
Fail
812
816
Quad request bits set in kr_spin?
Yes
Restore SPL and rrupts
No
Atomically set quad's bit
824
828
KQR_RELEASE set?
Yes
No
840
832
CPU request bits set in krq_spin?
Yes
set KRQ CP_HANDOFF
No
836
844
Atomically set CPU's hold bit
Restore rrupts
848
Succeed

FIG. 8

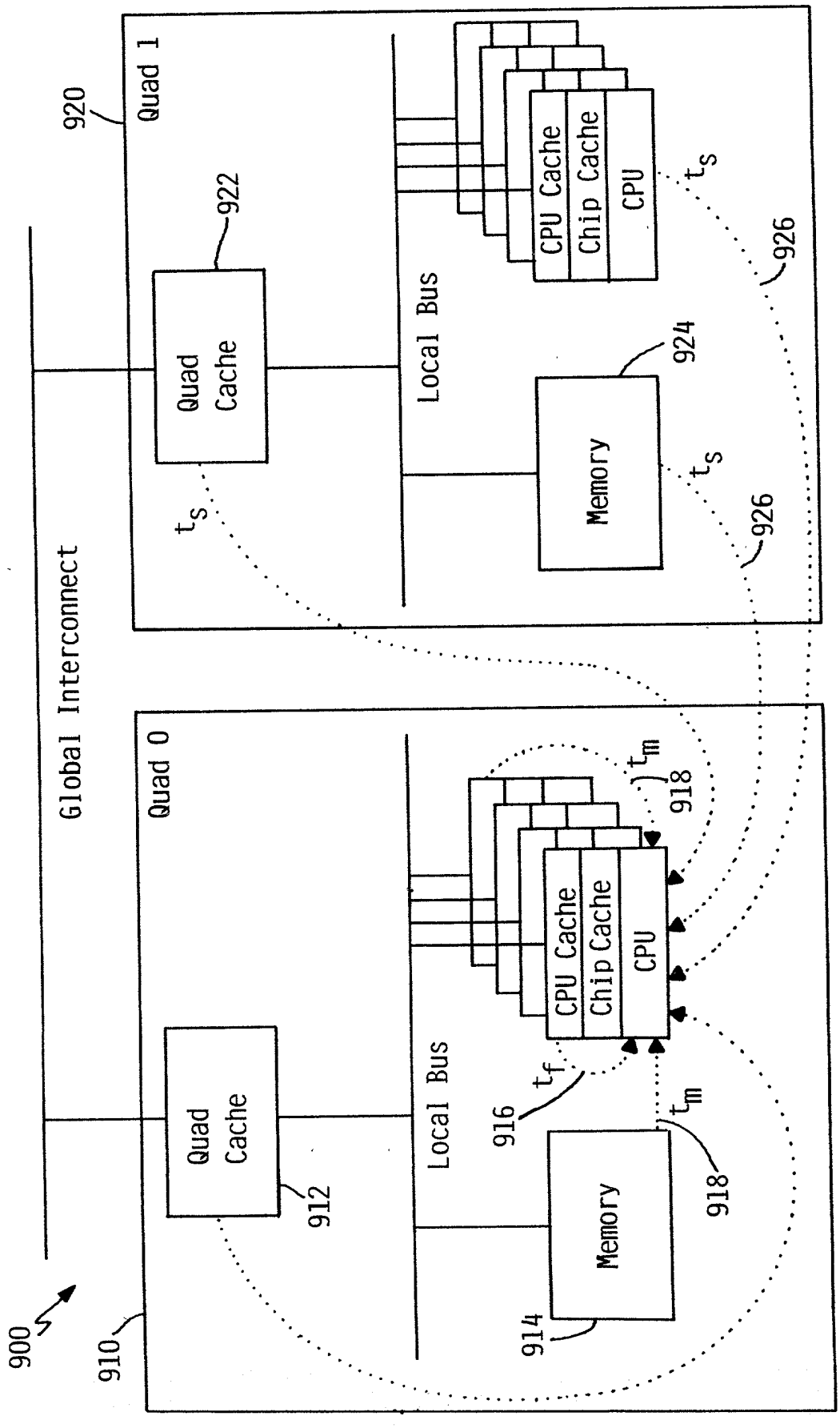

FIG. 9 (PRIOR ART)